Solving Simplification Ordering Constraints

Patricia Johann* and Rolf Socher-Ambrosius**

Abstract. This paper considers the decision problem for the existential fragment of the theory of simplification orderings. A simple, polynomial-time procedure for deciding satisfiability of such constraints is given, and it is also shown that the corresponding problem for the theory of total simplification orderings is NP-complete. This latter result can be interpreted as showing that the problem of deciding whether or not a simplification ordering on first-order terms can be linearized is NP-complete.

1 Introduction

The concept of well-founded ordering is ubiquitous in automated deduction, being fundamental to the development of termination proofs — especially for term rewriting systems and other rewriting-like relations — and of deduction strategies aiming to restrict search spaces in saturation processes. In particular, it plays a central role in describing ordered rewriting, ordered resolution, and ordered superposition.

Determining the existence of critical pairs in ordered completion processes requires, and in some cases corresponds precisely to, deciding solvability of ordering constraints. In such applications, ordering constraints act as a filter prohibiting completion steps allowable in an unordered setting but inconsistent with whatever deduction strategy the particular constraints enforce. To determine that admissible completion inferences remain available, solvability of the relevant constraints must be established.

The basic constraint satisfaction problem is that of deciding, given a signature Σ and a system

$$\Gamma \equiv \langle s_1 > t_1 \rangle, ..., \langle s_n > t_n \rangle$$

of term-pairs over Σ, whether or not there exist a ground substitution σ and a well-founded ordering $\succ$ such that $s_i\sigma \succ t_i\sigma$ holds for $i = 1, ..., n$. Depending on Σ, two subproblems can be distinguished: *constraint solving over fixed signatures*, in which σ is a Σ-substitution and $\succ$ is an ordering on Σ-terms, and *constraint solving over extended signatures*, where σ and $\succ$ may range over arbitrary extensions of Σ. Significantly, the same system of term-pairs over Σ may manifest vastly different behavior according as it is considered over fixed or extended signatures. Given the

* Fachbereich Informatik, Universität Saarbrücken, D-66123 Saarbrücken, Germany, *pjohann@cs.uni-sb.de*. On leave from the Department of Mathematics and Computer Science, Hobart and William Smith Colleges, Geneva, NY 14456, (315) 789-5500, *johann@hws.bitnet*.

** Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany, (0681) 302 5367, *socher@mpi-sb.mpg.de*. Supported by the German Ministry for Research and Technology (BMFT) under grant ITS 9103.

signature consisting of a single constant a, for example, the system $\Gamma \equiv \langle x, a \rangle$ is unsolvable when the signature is considered fixed. But over extended signatures, a new constant b may be introduced, so that the substitution $\{x \mapsto b\}$, together with any ordering $\succ$ such that $a \succ b$, is a solution.

Each specification of a term algebra and an ordering or class of orderings gives rise to an instance of the constraint satisfaction problem. Typically, the ordering $\succ$ is specified as part of a given problem (so that only an appropriate substitution σ is sought), and the purely existential fragment of the resulting theory is considered (two exceptions are the studies of the Σ_3 fragment of the theory of the subterm ordering and the Σ_4 fragment of the theory of an arbitrary but fixed partial recursive path ordering comprising [Ven87] and [Tra90], respectively). Depending on the intended application, $\succ$ has been interpreted variously as any recursive path ordering (RPO) or lexicographic path ordering (LPO) based on a fixed precedence, as the subterm relation, and as the embedding relation. The satisfiability problems for the existential fragments of the most commonly considered of these theories, namely those for fixed RPOs and LPOs, are known to be decidable ([JO91], [Com90]). Venkataraman ([Ven87]) has shown that the decision problem for the existential fragment of the theory of the subterm ordering is also decidable; recently Boudet and Comon ([BC93]) proved the corresponding result for the positive existential fragment of the embedding relation. All of these decision problems are likely NP-complete.

Although we treat a more general problem in Section 2.3, in this paper we focus on the simplification ordering (SO) constraint satisfaction problem. That is, we do not specify $\succ$ as part of our constraint satisfaction problems except to require that it be some arbitrary SO. Efficient techniques for deciding satisfiability of SO constraints have recently been employed to prune completion search spaces associated with theorem proving using ordering constrained clauses ([NR92]). In such settings, clauses with unsatisfiable ordering constraints may be deleted without sacrificing the refutational completeness of the resulting saturated sets of clauses.

Decidability of the SO constraint satisfaction problem over extended signatures is proved in [Pla93] by means of rather complicated techniques also establishing polynomial-time decidability of termination of ground term rewriting systems. We present in Section 2.2 a simple algorithm of the same low polynomial-time complexity as Plaisted's for deciding solvability of SO constraint problems over extended signatures, although we do not address here the apparently much harder problem of deciding solvability of SO constraint problems over fixed signatures. We further show for the first time that if $\succ$ is required to be total then the decidability of ground SO constraint satisfaction is NP-complete (so that deciding satisfiability over both fixed and extended signatures is at least as hard), and derive as a corollary to the proof the fact that the problem of deciding constraint satisfaction for an arbitrary LPO is NP-complete. In light of the polynomial-time complexity of deciding SO constraint satisfaction, this result can be interpreted as showing that the problem of deciding whether or not a given SO on first-order terms over a signature can be linearized is NP-complete.

That solvability of SO constraints can be decided quickly represents the main advantage of using them, rather than more structured and specific constraints, in automated deduction. Indeed, our algorithm for extended signatures executes in a time proportional to the cube of the number of symbol occurrences in the input problem (the complexity of the SO constraint satisfaction problem over fixed signatures is not yet clear). On the other hand, SO constraint solving techniques handle many orderings simultaneously and so admit rewrite or completion steps if there is *any* SO permitting them, and as a result, longer sequences are possible than for RPOs and LPOs. Nonetheless, the efficiency of deciding solvability of SO constraint problems (at least over extended signatures) may in practice more than compensate for the added generality in the orderings considered and, as Plaisted observes, SO constraint solving techniques may prove useful for theorem proving applications which require the existence of a single SO consistent with all choices (as evidenced by the constraints currently under consideration) made thus far in a particular theorem proving session or rewrite sequence.

A second argument in favor of choosing SO constraints over RPO or LPO constraints in automated deduction applications is their utility in proving termination of term rewriting systems. Termination of term rewriting systems is undecidable in general, even under very restrictive assumptions on the system. A sufficient condition for a term rewriting system R to be terminating is the existence of an SO $\succ$ such that $s\sigma \succ t\sigma$ holds for all ground instances $s\sigma \to t\sigma$ of rules $s \to t$ in R. It is indeed possible that this condition is satisfied even though there exists no RPO or LPO with the required property. In such cases, the question, "Is R terminating?" can be answered in the affirmative by appealing to more general SOs.

2 Constraint Solving in Extended Signatures

Although in this paper we focus primarily on SO constraint satisfaction problems comprising pairs of the form $\langle s > t \rangle$, in Section 2.3 we also consider problems involving somewhat more general types of constraint pairs. Section 2.2 treats the basic SO constraint satisfaction problem discussed in the introduction. We emphasize that we are only interested in deciding SO constraint satisfaction, as opposed to finding explicit solutions for these constraint problems.

2.1 The Basic Problem

We briefly recall the basic notation used in the text. For notations not explicitly given here the reader is referred to [DJ90].

A *signature* is a finite set Σ of (ranked) function symbols. For each $k \geq 0$, the set of function symbols in Σ of rank k is denoted Σ_k; the elements of Σ_0 are called *constants*. A signature Σ' is an *extension* of the signature Σ if $\Sigma \subseteq \Sigma'$, and a *proper extension* of Σ if Σ is a proper subset of Σ'.

Fix a countably infinite set V of variables. For each signature Σ, the set of *terms (over Σ)* is the smallest algebra $T(\Sigma, V)$ containing V and closed under the term formation rule: from $f \in \Sigma_k$ and $t_1, ..., t_k \in T(\Sigma, V)$ infer $f(t_1, ..., t_k) \in T(\Sigma, V)$. We write $V(t)$ for the set of variables occurring in the term t, and denote by $T(\Sigma)$ the subalgebra of *ground terms (over Σ)*, *i.e.*, of terms $t \in T(\Sigma, V)$ such that $V(t) = \emptyset$. Syntactic identity between terms is denoted by $\equiv$.

A SO *(over Σ)* is an ordering $\succ$ on the algebra $T(\Sigma)$ of ground terms over Σ satisfying

– *i)* the *subterm* property $f(...s...) \succ s$ and
– *ii)* the *monotonicity* property $f(...s...) \succ f(...t...)$ whenever $s \succ t$

for all $f(...s...), f(...t...) \in T(\Sigma)$. Note that SOs need not be total.

We will use transformation-based methods to solve SO constraint problems and so will represent such problems, as well as their solutions, in terms of constraint systems. Transformation-based methods attempt to reduce systems to so-called *solved systems* which represent their own solutions. We make these notions precise below.

A SO *constraint (over Σ)* is an ordered term-pair of the form $\langle s > t\rangle$ for $s, t \in T(\Sigma, V)$, and an *equational constraint (over Σ)* is an unordered term-pair of the form $\langle s = t\rangle$ for $s, t \in T(\Sigma, V)$. A *constraint (over Σ)* is either an SO constraint or an equational constraint over Σ; a *constraint system (over Σ)* is either the *trivial system* $\perp$ or a finite set of constraints over Σ. A constraint system is *pure* if all of its constraints are SO constraints. We write Γ, C rather than $\Gamma \cup \{C\}$ if Γ is a constraint system and C is a constraint. But because such a decomposition is ambiguous — Γ may or may not contain the constraint pair C — we write $\Gamma; C$ to indicate that C does not appear in Γ.

A term $t \in T(\Sigma, V)$ *occurs* in Γ if it is a component of a constraint in Γ; write $T(\Gamma)$ for the set of terms occurring in Γ. By $V(\Gamma)$ we denote $\bigcup\{V(t) \mid t \in T(\Gamma)\}$.

If Σ is any signature and t is any term in $T(\Sigma, V)$, denote by $[t]$ the (ordered) list $(t_1, ..., t_k)$ of immediate subterms of t. If Γ is a constraint system, write $(s_1, ..., s_k) >_\Gamma (t_1, ..., t_k)$ to indicate that for $i = 1, ..., k$ either $\langle s_i > t_i\rangle \in \Gamma$ or $s_i \equiv t_i$, and that $(s_1, ..., s_k) \not\equiv (t_1, ..., t_k)$.

A *substitution (over Σ)* is a finitely supported map from V into $T(\Sigma, V)$. Any substitution σ induces a map from $T(\Sigma, V)$ into itself in a natural way; we abuse notation and write σ for this extended map as well. If σ maps into $T(\Sigma)$ rather than all of $T(\Sigma, V)$ then σ is a *ground substitution (over Σ)*. Substitutions are postfixed, so that $t\sigma$ is the result of applying the substitution σ to the term t. By $\Gamma\sigma$ we denote the constraint system comprising all pairs $\langle s\sigma \, \rho \, t\sigma\rangle$ for $\langle s \, \rho \, t\rangle \in \Gamma$, $\rho \in \{>, =\}$.

Let Γ be a constraint system over Σ, σ be a ground substitution over Σ, and $\succ$ be an SO over Σ. The pair $\langle \sigma, \succ\rangle$ is a *solution for Γ (over Σ)* if $s\sigma \equiv t\sigma$ for all equational constraints $\langle s = t\rangle \in \Gamma$ and $s \succ t$ for all SO constraints $\langle s > t\rangle$ in Γ. We define the trivial constraint system $\perp$ to have no solutions.

Note that any constraint system over Σ is also a constraint system over Σ' for any extension Σ' of Σ, and likewise for SOs, substitutions, and hence solutions of Γ over Σ.

The following easy lemma shows that the processes of solving SO constraints and solving equational constraints can be effectively divorced.

Lemma 2.1 *Let Γ be a constraint system over a signature Σ whose* SO *constraints and equational constraints comprise subsystems Δ and Π, respectively. If σ is a most general unifier of Π, then Γ is satisfiable iff $\Delta\sigma$ is.*

We may therefore restrict attention to pure SO constraint problems when considering the basic SO constraint satisfaction problem. We describe in Section 2.2 a procedure for solving such constraint systems over possibly extended signatures. The question our procedure answers can be stated thusly: Given a signature Σ and a (pure) constraint system Γ over Σ, do there exist a ground substitution σ over some extension Σ' of Σ and an SO $\succ$ on $T(\Sigma')$ containing $\Gamma\sigma$?

2.2 The Procedure

We give below a set of transformations capable of transforming any (pure) constraint system Γ into a trivial or solved system and, in the latter case, determining a fragment of a SO on a finite set of terms containing appropriate instantiations of those in Γ. The key step in proving the correctness of our decision procedure for SO constraint satisfaction problems is then to see that any such fragment can be extended to a proper SO providing a solution for Γ.

Definition 2.2 The set $\mathcal{T}$ comprises the following transformations on constraint systems:

1. SUBTERM
$$\Gamma \Longrightarrow \Gamma; \langle s > t \rangle$$
if s occurs in Γ, $t \in [s]$, and $\langle s > t \rangle \notin \Gamma$.
2. TRANS
$$\Gamma \Longrightarrow \Gamma; \langle s > t \rangle$$
if there exist terms s, t and u such that $\langle s > u \rangle, \langle u > t \rangle \in \Gamma$ and $\langle s > t \rangle \notin \Gamma$.
3. LIFT
$$\Gamma \Longrightarrow \Gamma; \langle s > t \rangle$$
if s and t both occur in Γ, $[s] >_\Gamma [t]$, and $\langle s > t \rangle \notin \Gamma$.

Theorem 2.3 *Every sequence of $\mathcal{T}$-reductions terminates.*

Proof. Let $\mu(\Gamma)$ be the number of (unordered) pairs (s, t) of distinct subterms of terms in $T(\Gamma)$ which do not appear in any constraint $\langle s > t \rangle$ in Γ. If $\Gamma \Longrightarrow \Gamma'$ by SUBTERM, TRANS, or LIFT, then $\mu(\Gamma') < \mu(\Gamma)$. □

It is easy to check that if $\Gamma \Longrightarrow \Gamma'$ by a transformation in $\mathcal{T}$, then Γ is a constraint system over Σ iff Γ' is. The soundness of the transformations in $\mathcal{T}$ is likewise immediate.

Lemma 2.4 *The transformations in $\mathcal{T}$ are sound,* i.e., *if Γ' is a constraint system over Σ, $\Gamma \Longrightarrow \Gamma'$, and $\langle \sigma, \succ \rangle$ is a solution over an extension Σ' of Σ for Γ', then $\langle \sigma, \succ \rangle$ is also a solution over Σ' for Γ.*

Definition 2.5 A constraint system Γ is *consistent* provided it contains no constraint of the form $\langle s > s \rangle$, and *inconsistent* otherwise. A constraint system is *solved* if it is consistent and irreducible by $\mathcal{T}$.

Every sequence of $\mathcal{T}$-reductions must, by definition, terminate in a constraint system which is either solved or inconsistent. It is clear that if Γ is any inconsistent constraint system over Σ and Σ' is any extension of Σ, then Γ has no solution over Σ'. In particular, an inconsistent constraint system Γ over Σ has no solution over Σ itself. The remainder of this subsection is devoted to showing that any solved constraint system has a solution over extended signatures.

Although an SO over Σ is by definition an ordering on the set $T(\Sigma)$ of *all* ground terms over Σ, in the following we will have occasion to talk about SO "fragments," *i.e.*, orderings which are SOs when restricted to some finite set of ground terms. Call an ordering $\succ$ on a finite set $S \subseteq T(\Sigma)$ a SO *on* S if *i)* $f(...s...) \succ s$ and *ii)* $s \succ t$ implies $f(...s...) \succ f(...t...)$ for all terms $f(...s...), f(...t...) \in S$. If $\succ$ is an SO on S and $S' \subseteq S$, then $\succ$ is an SO on S' as well.

A solved constraint system Γ determines a SO on (appropriate instantiations of) the set $T(\Gamma)$ of terms occurring in Γ in a natural way:

Lemma 2.6 *If Σ is a signature and Γ is a solved constraint system over Σ, then there exists an extension Σ' of Σ, a ground substitution σ over Σ', and an* SO $\succ_\Gamma$ *on $T(\Gamma\sigma)$ containing $\Gamma\sigma$.*

Proof. Let Γ be a solved constraint system over Σ, let c_x be a new constant for every $x \in V(\Gamma)$, and let Σ' be the signature $\Sigma \cup \{c_x \mid x \in V(\Gamma)\}$. If σ is the substitution with components $\{x \mapsto c_x\}$ for each $x \in V(\Gamma)$, then the constraint system $\Gamma\sigma$ is solved since Γ is, and induces a simplification ordering $\succ_\Gamma$ on $T(\Gamma\sigma)$. □

The next sequence of lemmas, culminating in Theorem 2.15, shows how — in the notation of Lemma 2.6 — the ordering $\succ_\Gamma$ can be extended to an SO on all of $T(\Sigma')$. The subterm and monotonicity properties given earlier in this section extend to arbitrary binary relations in a straightforward fashion.

Definition 2.7 Let $\succ$ be an SO on the finite subterm closed set $S \subseteq T(\Sigma)$, and suppose $u \in T(\Sigma) \setminus S$ is such that $[u] \subseteq S$. Write $\succ_u$ for the smallest transitive binary relation on $S \cup \{u\}$ containing $\succ$ and satisfying the subterm property and the monotonicity property. We call $\succ_u$ a *simple extension* of $\succ$ by u.

For any term u, $\succ$ is a subrelation of $\succ_u$. We would like to see that $\succ_u$ is in fact an SO. To prove this, it will be helpful to know that if $s, t \in S$ and $s \succ_u t$, then $s \succ t$. The next few lemmas are dedicated to obtaining this result.

Throughout the remainder of this subsection, let $\succ_u$ be a simple extension of the SO $\succ$ on a finite subterm closed set $S \subseteq T(\Sigma)$. Define a $\succ_u$*-proof (from* $\succ$*)* for $s \succ_u t$ to be a tree P constructed according to the rules *Subterm*, *Trans*, and *Lift* of the inference system $\mathcal{IT}$ obtained by replacing all occurrences of the formal symbol $>$ in transformations in $\mathcal{T}$ by the SO $\succ_u$ and regarding each transformation, after this modification, as an inference rule. The nodes of P are labelled with systems of judgements of the form $s \succ_u t$, while the leaves of P are labelled with systems of judgements of the form $s \succ_u t$ for which the judgements $s \succ t$ also hold. The tree P is constructed from its nodes by *proof steps*, *i.e.*, by applications of inference rules from $\mathcal{IT}$. A proof step p is *critical* if u occurs in some premise of p but does not appear in any of the pairs introduced by p.

Lemma 2.8 *If P is any $\succ_u$-proof, then any critical step of P is of the form*

$$\frac{\Gamma; \langle s \succ_u u\rangle; \langle u \succ_u t\rangle}{\Gamma, \langle s \succ_u u\rangle, \langle u \succ_u t\rangle, \langle s \succ_u t\rangle} \qquad (*)$$

Proof. By definition, a critical proof step cannot be a *Subterm* step. If the critical step p is a *Lift* step, then since u occurs in the premise of p, u must be a subterm of one of the terms introduced by p, contradicting the fact that that p is critical. Thus p must be a *Trans* step, and so it is easy to see that p must be of the form $(*)$. □

Definition 2.9 The proof transformation system $\mathcal{P}$ on $\succ_u$-proof trees is given by the rules of Figure 1.

$$\dfrac{\dfrac{s \succ t \qquad t \succ u}{s \succ u} \qquad u \succ v}{s \succ v} \quad\Longrightarrow\quad \dfrac{s \succ t \qquad \dfrac{t \succ u \qquad u \succ v}{t \succ v}}{s \succ v} \qquad (\mathcal{P}_1)$$

$$\dfrac{s \succ u \qquad \dfrac{u \succ t \qquad t \succ v}{u \succ v}}{s \succ v} \quad\Longrightarrow\quad \dfrac{\dfrac{s \succ u \qquad u \succ t}{s \succ t} \qquad t \succ v}{s \succ v} \qquad (\mathcal{P}_2)$$

$$\dfrac{\dfrac{[s] \succ [u]}{s \succ u} \qquad u \succ t}{s \succ t} \quad\Longrightarrow\quad \dfrac{s \succ v \qquad v \succ t}{s \succ t} \qquad (\mathcal{P}_3)$$

if $t \in [u], t \succ v,$ and $v \in [s]$.

$$\dfrac{\dfrac{[s] \succ [u]}{s \succ u} \qquad u \succ t}{s \succ t} \quad\Longrightarrow\quad \dfrac{}{s \succ t} \qquad (\mathcal{P}_4)$$

if $t \in [u]$ and $t \in [s]$.

$$\dfrac{\dfrac{[s] \succ [u]}{s \succ u} \qquad \dfrac{[u] \succ [t]}{u \succ t}}{s \succ t} \quad\Longrightarrow\quad \dfrac{\dfrac{[s] \succ [u] \qquad [u] \succ [t]}{[s] \succ [t]}}{s \succ t} \qquad (\mathcal{P}_5)$$

Fig. 1. The proof transformation system $\mathcal{P}$.

Lemma 2.10 *Every sequence of applications of rules from $\mathcal{P}$ terminates.*

Proof. Each application of a rule from $\mathcal{P}$ to a $\succ_u$-proof P decreases the number of occurrences of u in the tree P. □

Lemma 2.11 *If the $\succ_u$-proof P is irreducible under the transformation system $\mathcal{P}$, then P does not contain any critical steps.*

Proof. Let P be irreducible under $\mathcal{P}$ and suppose contains a critical step, *i.e.*, a step of the form $(*)$. The inferences $s \succ_u u$ and $u \succ_u t$ must both be conclusions of the inference rules *Subterm*, *Trans*, and *Lift* in $\mathcal{IT}$; we derive a contradiction by cases according to the rules of which they are the conclusions. If $s \succ_u u$ is the conclusion of *Trans*, then P is reducible by $\mathcal{P}_1$, and if $u \succ_u t$ is the conclusion of *Trans*, then P is reducible by $\mathcal{P}_2$. Since $s \succ_u u$ cannot be the conclusion of *Subterm*, the only other possibility is that it is the conclusion of *Lift*. Since $u \succ_u t$ must be the conclusion of either *Lift* or *Subterm*, two cases obtain. In the first, P is reducible by $\mathcal{P}_5$. In the second, t must be an element of $[u]$ and, since P contains the inference $[s] \succ_u [u]$, either t is an element of $[s]$ so that $\mathcal{P}_4$ applies to P, or else there is a subterm s' of s such that $s' \succ_u t$ so that $\mathcal{P}_3$ applies to P. Since these considerations exhaust all possibilities, we conclude that P, being irreducible under $\mathcal{P}$, can contain no critical step. □

Lemma 2.12 *If $\succ$ is an* SO *on a finite subterm closed set $S \subseteq T(\Sigma)$ and $s, t \in S$, then $s \succ_u t$ implies $s \succ t$.*

Proof. We show that if $s, t \in T(\Sigma)$ are such that $s \not\equiv u$, $t \not\equiv u$, and there exists a $\succ_u$-proof from $\succ$ of $s \succ_u t$, then there exists a $\succ_u$-proof from $\succ$ of $s \succ_u t$ in which u does not appear. The result then follows easily.

If P is a $\succ_u$-proof from $\succ$ of $s \succ_u t$, then since $\mathcal{P}$ is terminating, there must exist a $\succ_u$-proof P' from $\succ$ of $s \succ_u t$ which is irreducible by $\mathcal{P}$. By Lemma 2.11, P' can contain no critical steps, so that $s \not\equiv u$ and $t \not\equiv u$ together imply that P' does not contain any occurrence of u. □

It is now a simple matter to see that $\succ_u$ is an SO:

Lemma 2.13 *Let $\succ$ be a (total)* SO *on a finite subterm closed set $S \subseteq T(\Sigma)$ and let $u \in T(\Sigma) \setminus S$ be such that $[u] \subseteq S$. Then $\succ_u$ is a (total)* SO *on $S \cup \{u\}$.*

Proof. Let $\succ_u$ be the simple extension of $\succ$ by the term u, and assume to the contrary that $\succ_u$ is not an SO. Since $\succ_u$ satisfies the subterm, transitivity, and monotonicity properties, the only way $\succ_u$ can fail to be a SO is if there exists a term s such that $s \succ_u s$.

If $s \not\equiv u$, then Lemma 2.12 implies that $s \succ s$, which contradicts the fact that $\succ$ is an SO. So suppose $u \succ u$ and let P be a $\succ_u$-proof from $\succ$ of $u \succ_u u$. Clearly $u \succ_u u$ cannot be the conclusion of *Subterm*. If it is the conclusion of *Lift*, then there is an immediate subterm u' of u such that $u' \succ u'$. Since $u' \not\equiv u$, Lemma 2.12 implies that $u' \succ u'$, again contradicting the fact that $\succ$ is a SO. Finally, if $u \succ_u u$ is the conclusion of *Trans*, then there exists a term $t \not\equiv u$ such that $u \succ_u t$ and $t \succ_u u$. Then since $\succ$ satisfies the transitivity property, we must have $t \succ_u t$ and hence $t \succ t$, so that this case is dispensed with as above.

Finally, if $\succ$ is total, then the smallest ordering containing $\succ_u$ and the relation $t \succ_u u$, where t is the immediate successor of the least upper bound of the elements of $[u]$ with respect to $\succ$, and satisfying the transitivity and monotonicity properties, is a total SO on $S \cup \{u\}$. □

We now see how repeated simple extensions of an SO on a finite set of ground terms can be used to specify a SO on the set of all ground terms.

Lemma 2.14 *Let $\succ$ be a (total)* SO *on a finite subterm closed set $S \subseteq T(\Sigma)$. Then $\succ$ can be extended to a (total)* SO *on $T(\Sigma)$.*

Proof. We can enumerate the elements of $T(\Sigma) \setminus S$ in a form $t_1, t_2, ...$ such that $[t_i] \subseteq S \cup \{t_1, ..., t_{i-1}\}$. Define a sequence $\succ_0, \succ_1, ...$ of SOs such that $\succ_0$ is precisely $\succ$ and each $\succ_i$, $i > 0$, is the simple extension by t_i of the SO $\succ_{i-1}$ on $S \cup \{t_1, ..., t_{i-1}\}$. Finally, define a relation $\succ'$ on $T(\Sigma)$ by declaring $s \succ' t$ to hold precisely when $s \succ_i t$ holds for some $i \geq 0$.

That $\succ'$ is indeed an SO can be verified as follows. If $s \succ' t$ and $t \succ' u$ for some $s, t, u \in T(\Sigma)$, then $s \succ_i t$ and $t \succ_j u$ hold for some $i, j \geq 0$. Thus $s \succ_k t$, $t \succ_k u$, and hence $s \succ_k u$ hold, where $k = max(i, j)$, and so $s \succ' u$ holds as well. Also, if $s \succ' s$ holds, then $s \succ_i s$ holds for some i, contradicting the fact that $\succ_i$ is an ordering. Thus $\succ'$ is itself an ordering. Similar argumentation establishes that $\succ'$ satisfies the subterm and monotonicity properties, and the proof that totality of $\succ$ implies totality of $\succ'$ is also straightforward. □

Theorem 2.15 *For any signature Σ, any solved constraint system over Σ has a solution over some extension of Σ.*

Proof. Let Γ be a solved constraint systems over Σ. Then by Lemma 2.6, there exist an extension Σ' of Σ, a ground substitution σ over Σ', and an SO $\succ_\Gamma$ on $T(\Gamma\sigma) \subseteq T(\Sigma')$. By Lemma 2.14, $\succ_\Gamma$ can be extended to an SO $\succ$ on all of $T(\Sigma')$ containing $\Gamma\sigma$. Thus $\langle \sigma, \succ \rangle$ is a solution over Σ' for Γ. □

The results of this section yield a simple procedure for solving SO constraint problems over extended signatures. Given a constraint system Γ over Σ, we may obtain a $\mathcal{T}$-irreducible system by repeated application of inference rules from $\mathcal{T}$. If the resulting system Γ' is inconsistent, then it has no solution (over fixed or extended signatures). But if Γ' is in fact consistent, then it is by definition solved, and Theorem 2.15 guarantees the existence of a solution $\langle \sigma, \succ \rangle$ for Γ' over some extension Σ' of Σ. By repeated application of Lemma 2.4 we see that $\langle \sigma, \succ \rangle$ is also a solution over Σ' for Γ.

2.3 A More General Problem

In this section we sketch some modifications that can be made to the procedure of Section 2.2 to accommodate constraint systems containing pairs of the form

$$\langle s \geq t \rangle, \langle s \not> t \rangle, \langle s \not\geq t \rangle, \text{ and } \langle s \neq t \rangle$$

in addition to those of the form $\langle s = t \rangle$ and $\langle s > t \rangle$ considered there. Although constraints of the form $\langle s \geq t \rangle$ can ostensibly be handled by trading them for disjunctive constraint systems of the form $\langle s > t\rangle \vee \langle s = t\rangle$ and then transforming the resulting systems into disjunctive normal forms, such a naive approach can lead to constraint systems which grow exponentially. More importantly, it is not possible to reduce constraints of the form $\langle s \not> t\rangle$ or $\langle s \not\geq t\rangle$ to constraints involving only the equality and SO constraints of Section 2.2 if we do not require that solution orderings for SO constraint satisfaction problems be total.

We will make use of the following terminology in this subsection. An equational constraint $\langle s = t\rangle \in \Gamma$ is said to be *solved in* Γ if (at least) one of s and t is a variable occurring only once in Γ. If $\langle x = t\rangle$ is solved in Γ, we say that x is a *solved variable* of Γ. Although any pair $\langle x, y\rangle$ both of whose component variables appear only once in Γ determines two solved variables, we will assume that there is a uniform way of choosing which one will be considered solved, and we will consider the other to be unsolved. Any substitution θ then determines a system of solved equational constraints in a natural way; we write $[\theta]$ for such a constraint system.

A *solution (over* Σ*)* for a generalized constraint system Γ over Σ is a pair $\langle\sigma, \succ\rangle$, where σ is a ground substitution over Σ and $\succ$ is an SO over Σ such that

- $s\sigma \equiv t\sigma$ holds for each pair $\langle s = t\rangle \in \Gamma$,
- $s\sigma \succ t\sigma$ holds for each pair $\langle s > t\rangle \in \Gamma$,
- $s\sigma \succ t\sigma$ or $s\sigma \equiv t\sigma$ holds for each pair $\langle s \geq t\rangle \in \Gamma$,
- $s\sigma \not\equiv t\sigma$ holds for each pair $\langle s \neq t\rangle \in \Gamma$,
- $s\sigma \succ t\sigma$ does not hold for each pair $\langle s \not> t\rangle \in \Gamma$, and
- neither $s\sigma \succ t\sigma$ nor $s\sigma \equiv t\sigma$ hold for each pair $\langle s \not\geq t\rangle \in \Gamma$.

To appropriately modify the set $\mathcal{T}$ of transformations for this more general setting, we begin by defining a total ordering on the constraint symbols $>, \geq$, and $=$ by declaring $>$ to be strictly larger than $\geq$, which is in turn strictly larger than $=$. If Γ is a (generalized) constraint system over a signature Σ, then write $(s_1, ..., s_k)\, \rho_\Gamma\, (t_1, ..., t_k)$ to indicate that

- $\rho_i \in \{>, \geq, =\}$ for $i = 1, ..., k$,
- for some $i \in \{1, ..., k\}$, ρ_i is different from $=$,
- $(s_1, ..., s_k) \not\equiv (t_1, ..., t_k)$, and no solved variables in Γ appear in these lists,
- either $\langle s_i\, \rho_i\, t_i\rangle \in \Gamma$ or $s_i \equiv t_i$ for $i = 1, ..., k$, and
- $\rho_\Gamma = max\{\rho_i \mid i = 1, ..., k\}$.

For example, if $\langle a > b\rangle$, $\langle c \geq d\rangle$, and $\langle e = f\rangle$ are in Γ, then $(a, c) >_\Gamma (b, d)$, $(c, e) \geq_\Gamma (d, f)$, and $(a, e) >_\Gamma (b, f)$.

The transformation SUBTERM requires no modification in the more general setting, but TRANS requires an additional clause stating that if $\langle s\, \rho_1\, u\rangle$ and $\langle u\, \rho_2\, t\rangle$ are in Γ, for $\rho_1, \rho_2 \in \{>, \geq\}$, but $\langle s\, \rho_\Gamma\, t\rangle$ is not, then $\langle s\, \rho_\Gamma\, t\rangle$ can be added to Γ. Finally, all occurrences of "$>$" and "$>_\Gamma$" in LIFT must be replaced by "ρ_Γ."

By contrast with Lemma 2.1, we cannot separate equality and SO constraints in the generalized setting, due to the possibility of "cycling." For example, ordering constraints $\langle s \geq t\rangle$ and $\langle t \geq s\rangle$ require that $s\sigma \equiv t\sigma$ for any solution $\langle\sigma, \succ\rangle$ for a constraint system containing them. Moreover, certain combinations of constraints in a system Γ, *e.g.*, $\langle s > t\rangle$ and $\langle s \not\geq t\rangle$, guarantee that Γ will have no solution. We therefore introduce “bookkeeping” transformations

$$\begin{array}{ll} \Gamma; \langle s > s \rangle & \Longrightarrow \bot, \\ \Gamma; \langle s \neq s \rangle & \Longrightarrow \bot, \\ \Gamma; \langle s \not\geq s \rangle & \Longrightarrow \bot, \\ \Gamma; \langle s > t \rangle; \langle s \not\geq t \rangle & \Longrightarrow \bot, \\ \Gamma; \langle s \geq t \rangle; \langle t > s \rangle & \Longrightarrow \bot, \\ \Gamma; \langle s \, \rho \, t \rangle; \langle s \not\rho \, t \rangle & \Longrightarrow \bot, \end{array}$$

where ρ is any constraint symbol. Together with $\mathcal{T}$ and the transformations DELETE, CYCLE, CLASH and UNIFY given below, these comprise the set $\mathcal{GT}$.

The transformation DELETE is given by

$$\Gamma; \langle s \geq s \rangle \Longrightarrow \Gamma,$$

while CYCLE is defined by

$$\Gamma \Longrightarrow \Gamma; \langle s = t \rangle,$$

where $\langle s \geq t \rangle$ and $\langle t \geq s \rangle$, or $\langle s \geq t \rangle$ and $\langle s \not> t \rangle$, are both in Γ, and $\langle s = t \rangle$ is not in Γ. UNIFY is given by

$$\Gamma; \langle s = t \rangle \Longrightarrow \Gamma\theta, [\theta],$$

where θ is a most general unifier of s and t and $\langle s = t \rangle$ is not a solved pair in Γ. And CLASH is given by

$$\Gamma; \langle s = t \rangle \Longrightarrow \bot,$$

where s and t are not unifiable.

It is not hard to see that every sequence of $\mathcal{GT}$-reductions terminates. Let $\mu = \langle \mu_1, \mu_2, \mu_3, \mu_4, \mu_5 \rangle$ be the well-founded lexicographic ordering on constraint systems where $\mu_1(\Gamma)$ is the number of unsolved variables in $V(\Gamma)$, $\mu_2(\Gamma)$ is the number of (unordered) pairs (s, t) of distinct subterms of terms in $T(\Gamma)$ not appearing in any constraint $\langle s = t \rangle$, $\mu_3(\Gamma)$ is the number of (unordered) pairs (s, t) of distinct subterms of terms in $T(\Gamma)$ not appearing in any constraint $\langle s > t \rangle$ in Γ, $\mu_4(\Gamma)$ is the number of (unordered) pairs of distinct subterms (s, t) of terms in $T(\Gamma)$ not appearing in any constraint $\langle s \geq t \rangle$ in Γ, and $\mu_5(\Gamma)$ is the number of pairs in Γ. Then

- SUBTERM does not increase μ_1 or μ_2 and decreases μ_3,
- TRANS and LIFT do not increase μ_1 or μ_2, and either decrease μ_3 or else do not increase μ_3 and decrease μ_4,
- CYCLE does not increase μ_1 and decreases μ_2,
- UNIFY decreases μ_1, and
- DELETE, CLASH, and the bookkeeping transformations do not increase μ_1, μ_2, μ_3, or μ_4, and decrease μ_5.

The $\mathcal{GT}$ transformations clearly preserve solutions, in the sense that if $\Gamma \Longrightarrow \Gamma'$ then $\langle \sigma, \succ \rangle$ is a solution over Σ for Γ' iff it is a solution over Σ for Γ.

A constraint system is said to be *solved* if it is $\mathcal{GT}$-irreducible and non-trivial. Any sequence of $\mathcal{GT}$-reductions out of a constraint system Γ necessarily terminates in either the trivial system or in a solved system. If the resulting $\mathcal{GT}$-irreducible system is trivial, then by the previous paragraph, Γ has no solution. If the resulting $\mathcal{GT}$-irreducible system is solved, then we would like to see that it, and therefore Γ, has a solution. This is precisely the upshot of the next theorem.

Theorem 2.16 *Any $\mathcal{GT}$-irreducible and non-trivial constraint system over a signature Σ has a solution over some extension of Σ.*

Proof. Let Γ be a $\mathcal{GT}$-irreducible and non-trivial constraint system over Σ, let τ be a substitution assigning to each unsolved variable x in Γ a distinct constant c_x, let $\Sigma' = \Sigma \cup \{c_x \mid x \text{ is unsolved in } \Gamma\}$, and let

$$\sigma = \tau \cup \{x \mapsto t\tau \mid \langle x = t\rangle \in \Gamma\}.$$

Note that Γ is solved iff $\Gamma\sigma$ is, and that for any term s which is not a solved variable of Γ, $s\sigma \equiv s\tau$. Denote by $\succ_\Gamma$ the smallest binary relation containing $s\sigma \succ t\sigma$ for all pairs of terms occurring in constraints $\langle s\sigma > t\sigma\rangle$ or $\langle s\sigma \geq t\sigma\rangle$ in $\Gamma\sigma$, and satisfying the subterm, monotonicity, and transitivity properties. Then since $\Gamma\sigma$ is solved, $\succ_\Gamma$ is an ordering on the terms occurring in pairs $\langle s\sigma > t\sigma\rangle$ and $\langle s\sigma \geq t\sigma\rangle$ in $\Gamma\sigma$.

That $\succ_\Gamma$ is an SO on the terms occurring in pairs $\langle s\sigma > t\sigma\rangle$ and $\langle s\sigma \geq t\sigma\rangle$ in $\Gamma\sigma$ follows from the fact that $\Gamma\sigma$ is solved and therefore $\mathcal{GT}$-irreducible.

Now, since $\succ_\Gamma$ is an SO on the terms occurring in pairs $\langle s\sigma > t\sigma\rangle$ and $\langle s\sigma \geq t\sigma\rangle$ in $\Gamma\sigma$, by the results of Section 2.2, $\succ_\Gamma$ can be extended to a SO $\succ$ on all of $\mathcal{T}(\Sigma')$. We need only show that $\langle\sigma, \succ\rangle$ is a solution over Σ' for Γ.

- $s\sigma \succ t\sigma$ holds by definition of $\succ$ if $\langle s > t\rangle$ or $\langle s \geq t\rangle$ is in Γ.
- if $\langle s = t\rangle \in \Gamma$, then one of s and t is a solved variable in Γ and the other is a term none of whose variables are solved in Γ. If $s \equiv x$ for some solved variable x, then $t\sigma \equiv t\tau$, and therefore $s\sigma \equiv t\tau \equiv t\sigma$.
- $\langle s \not> t\rangle \in \Gamma$ (resp., $\langle s \not\geq t\rangle \in \Gamma$), then since Γ is $\mathcal{GT}$-irreducible, $\langle s\sigma > t\sigma\rangle \notin \Gamma\sigma$ (resp., $\langle s\sigma \geq t\sigma\rangle \notin \Gamma\sigma$). Thus $s\sigma \succ_\Gamma t\sigma$ cannot hold. On the other hand, $s\sigma \succ t\sigma$ cannot hold by definition of $\succ$ by Lemma 2.12.
- if $\langle s \neq t\rangle \in \Gamma$, then since Γ is $\mathcal{GT}$-irreducible, $s \equiv t$ cannot hold. By definition of σ, then, $s\sigma \not\equiv t\sigma$.

□

3 Complexity

In [Pla93], Plaisted shows that a procedure similar to ours for deciding satisfiability of (basic) SO constraint problems over extended signatures is of complexity $O(n^3)$, n being the total number of function symbol and variable occurrences in the input constraint system. Given his proof, it is not hard to see that our procedure has precisely the same time complexity.

On the other hand, Plaisted does not at all consider the behavior of SO constraint satisfaction problems when the SOs over which they are to solved are required to be total. We prove below that the problem of deciding satisfiability of ground constraints by total SOs, whether over fixed or extended signatures, is NP-complete.

Theorem 3.1 *The following problem (Total* SO *Constraint Solving) is* NP-*complete: Given a signature Σ and a ground constraint system*

$$\Gamma \equiv \langle s_1 > t_1\rangle, ..., \langle s_n > t_n\rangle$$

over Σ, does there exist a total SO $\succ$ *over Σ containing Γ?*

Proof. The problem is easily seen to be in NP. A non-deterministic algorithm need only guess an enumeration $u_1, u_2, ..., u_m$ of the set of all subterms of terms in $T(\Gamma)$. Then by the results of Section 2.2, satisfiability of the ground system

$$\Gamma' \equiv \langle u_1 > u_2 \rangle, \langle u_2 > u_3 \rangle, ..., \langle u_{m-1} > u_m \rangle$$

can be determined in polynomial time. If Γ' is satisfiable and Γ is a subsystem of Γ', then the results of Section 2.2 guarantee that Γ' induces a total SO $\succ$ on $T(\Sigma)$ containing Γ. Otherwise, the SO on $T(\Sigma)$ induced by Γ' does not provide a solution for Γ.

The rest of the proof of NP-completeness is achieved by a reduction of "Monotone 3-SAT" ([GJ79]), which is the problem of deciding satisfiability of a set $\mathcal{C}$ of propositional clauses of length three such that every clause C in $\mathcal{C}$ is positive (*i.e.*, has only positive occurrences of propositional symbols) or negative (*i.e.*, has only negative occurrences of propositional symbols).

The following table gives a transformation from a clause set $\mathcal{C}$ satisfying the assumptions of Monotone 3-SAT over the set $\mathcal{X}$ of propositional variables into a constraint system Γ over the signature $\Sigma = \mathcal{X} \cup \{f, g, 1\}$, where the elements of $\mathcal{X}$ are regarded as constants, f and g are ternary function symbols, and 1 is also a constant:

$$\neg x \vee \neg y \vee \neg z \implies f(1,1,1) > f(x,y,z)$$
$$x \vee y \vee z \implies g(x,y,z) > g(1,1,1)$$

We prove that the clause set $\mathcal{C}$ is satisfiable iff the constraint system Γ is satisfiable by some total SO.

First, suppose that Γ is satisfiable, *i.e.*, that there exists a total SO $\succ$ over Σ containing Γ. Since $\succ$ is total, either $x \succ 1$ or $1 \succ x$ holds for every $x \in \mathcal{X}$. Define

$$\mathcal{I} = \{x \in \mathcal{X} \mid x \succ 1\}.$$

We show that $\mathcal{I}$ is an interpretation satisfying $\mathcal{C}$, *i.e.*, that if $C \in \mathcal{C}$ is positive then it contains at least one variable in $\mathcal{I}$, and that if $C \in \mathcal{C}$ is negative, then it contains at least one variable not in $\mathcal{I}$. If $C = x \vee y \vee z$ is positive, then $g(x,y,z) \succ g(1,1,1)$. If x, y, and z are all not in $\mathcal{I}$, then from $1 \succ x$, $1 \succ y$, and $1 \succ z$ it follows that $g(1,1,1) \succ g(x,y,z)$, which is a contradiction. The case when C is negative is similar.

On the other hand, suppose that $\mathcal{C}$ is satisfiable by some interpretation $\mathcal{I} \subseteq \mathcal{X}$. Then every positive clause C in $\mathcal{C}$ contains an atom $x \in \mathcal{I}$. We reorder each clause C in such a way that x is the leftmost variable of C, *i.e.*, that $C = x \vee y \vee z$ with $x \in \mathcal{I}$. Similarly, if D is a negative clause, we write D in the form $D = \neg x \vee \neg y \vee \neg z$ with $x \notin \mathcal{I}$.

Now let $\succ$ be a lexicographic path ordering over Σ based on any total precedence ordering $\succ'$ satisfying

- $f \succ' x \succ' 1$ for all $x \in \mathcal{I}$, and
- $g \succ' 1 \succ' y$ for all $y \notin \mathcal{I}$.

Such a precedence clearly exists, and it is easy to verify that for all $C \in \mathcal{C}$ the ordering $\succ$ satisfies the constraint system corresponding to C. Moreover, any LPO based on a total precedence is itself total on ground terms. Thus we have seen that every Monotone 3-SAT problem can be polynomially reduced to a total SO constraint satisfaction problem, and this proves the SO constraint satisfaction problem to be NP-complete. □

In light of the polynomial time complexity of the constraint satisfaction problem for arbitrary SOs, Theorem 3.1 can be interpreted as showing that the problem of deciding whether or not a given SO on a set $T(\Sigma)$ of terms can be *linearized* — *i.e.*, embedded into a total SO on $T(\Sigma)$ — is NP-complete.

The proof of Theorem 3.1 also shows that the following similar problem (LPO *Constraint Solving*) is NP-complete: Given a signature Σ and a constraint system Γ over Σ, does there exist a precedence on Σ and an LPO $\succ$ over this precedence containing Γ? Note that this problem differs from the one treated in [Com90] and [NR92] in that in both of these papers an arbitrary but fixed precedence on Σ is assumed.

References

[BC93] A. Boudet and H. Comon. About the Theory of Tree Embedding. In Springer-Verlag LNCS 668, M.-C. Gaudel and J.-P. Jouannaud, eds., pp. 376 – 390, 1993.

[Com90] H. Comon. Solving Inequations in Term Algebras. In *Proceedings of the 5th IEEE Symposium on Logic in Computer Science*, pp. 62 – 29, 1990.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. In *Handbook of Theoretical Computer Science* B, J. van Leeuwen, ed., North-Holland, 1990.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP*-Completeness*. W. H. Freeman and Company, 1979.

[JO91] J.-P. Jouannaud and M. Okada. Satisfiability of Systems of Ordinal Notations with the Subterm Property is Decidable. In Springer-Verlag LNCS 510, J. L. Albert, B. Monien, and M. R. Artalejo, eds., pp. 455 – 468, 1991.

[NR92] R. Nieuwenhuis and A. Rubio. Theorem Proving with Ordering Constrained Clauses. In Springer-Verlag LNCS 607, D. Kapur, ed., pp. 477 – 491, 1992.

[Pla93] D. A. Plaisted. Polynomial Time Termination and Constraint Satisfaction Tests. In Springer-Verlag LNCS 690, C. Kirchner, ed., pp. 405 – 420, 1993.

[Tra90] R. Treinen. A New Method for Undecidability Proofs for First-order Theories. Technical Report A-09/90, Universität des Saarlandes, 1990.

[Ven87] K. N. Venkataraman. Decidability of the Purely Existential Fragment of the Theory of Term Algebras. *JACM* 34, pp. 492 – 510, 1987.